Install and Configure LibreNMS on RHEL/CentOS 8.x

Overview

- **LibreNMS** is an open-source auto-discovering network monitoring tool based on PHP, MySQL, and SNMP
- **LibreNMS** includes support for a wide range of network hardware and operating systems including Juniper, Cisco, Linux, Foundry, FreeBSD, Brocade, HP, Windows and many more. It is a fork of "Observium" monitoring tool

Features of LibreNMS:

- **Has Automatic discovery** – It will automatically discover your entire network using CDP, FDP, LLDP, OSPF, BGP, SNMP, and ARP
- **API Access** – LibreNMS provides a full API to manage, graph and retrieve data from your install.
- **Automatic Updates** – With LibreNMS you get to stay up to date automatically with new features and bug fixes.
- **Customizable alerting** – Highly flexible alerting system, notify via email, IRC, slack and more.
- **Support for Distributed Polling** through horizontal scaling which grows with your network
- **Billing system** – Easily generate bandwidth bills for ports on your network based on usage or transfer.
- **Android and iOS application** – There is a native iPhone/Android App is available which provides core functionality.
- **Multiple authentication methods:** MySQL, HTTP, LDAP, Radius, Active Directory

Configure SELinux in permissive mode:

```
# sed -i 's/SELINUX=.*/SELINUX=permissive/g' /etc/selinux/config
# setenforce 0
# getenforce
```

Step 1: Install dependencies:

Add EPEL repositories:

```
# dnf install epel-release
```

Install Development Tools and Other Required Packages:

```
# dnf groupinstall "Development Tools"
```

```
# dnf -y install yum-utils
```

Step 2: Install PHP and Apache

Install Apache Web server, PHP and extensions required by LibreNMS.

```
# dnf -y install https://rpms.remirepo.net/enterprise/remi-release-8.rpm
# dnf module reset php
# dnf module enable php:remi-8.1
# dnf install bash-completion cronie fping git httpd ImageMagick mariadb-server mtr net-snmp
 net-snmp-utils nmap php-fpm php-cli php-common php-curl php-gd php-json php-mbstring php-
 process php-snmp php-xml php-zip php-mysqlnd python3 python3-devel python3-PyMySQL python3-
 redis python3-memcached python3-pip python3-systemd rrdtool unzip
```

Configure and verify Timezone: Set PHP timezone inside /etc/php.ini file

```
# vi /etc/php.ini
 date.timezone = Asia/Dhaka
```

Remember to set the system timezone as well.

```
timedatectl set-timezone Asia/Dhaka
```

Enable NTP:

```
# timedatectl set-ntp true
```

Start and enable httpd and php-fpm services:

```
# systemctl enable --now httpd php-fpm
# systemctl start httpd
# systemctl start php-fpm
```

Confirm services status:

```
# systemctl status httpd php-fpm
```

Verify PHP version and loaded modules:

```
# php -v
# php -m
```

Allow http and https ports on the firewall:

```
# firewall-cmd --add-service={http,https} --permanent
# firewall-cmd --reload
```

Step 3: Install and Configure Database Server

MariaDB is a community-developed fork of the MySQL relational database management system.

```
# dnf install mariadb-server mariadb
```

Activate the mariadb service using the command below:

```
# systemctl enable --now mariadb
```

```
# systemctl start mariadb
# systemctl status mariadb
```

Run the command mysql_secure_installation to harden MariaDB database server security:

```
# mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!
In order to log into MariaDB to secure it, we'll need the current
password for the root user.  If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

Set root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
 ... Success!

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them.  This is intended only for testing, and to make the installation
go a bit smoother.  You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] y
 ... Success!

Normally, root should only be allowed to connect from 'localhost'.  This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] y
 ... Success!

By default, MariaDB comes with a database named 'test' that anyone can
access.  This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] y
 - Dropping test database...
 ... Success!
 - Removing privileges on test database...
 ... Success!
```

```
Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] y
 ... Success!

Cleaning up...

All done!  If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
```

After Database server installation, create database and user for LibreNMS Monitoring tool.

```
# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 16
Server version: 10.3.27-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE librenms CHARACTER SET utf8 COLLATE utf8_unicode_ci;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> CREATE USER 'librenms'@'localhost' IDENTIFIED BY 'LibreAdmin';
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON librenms.* TO 'librenms'@'localhost';
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> exit
```

Reconfigure /etc/my.cnf.d/mariadb-server.cnf file and add below lines within the [mysqld] section:

```
# vi /etc/my.cnf.d/mariadb-server.cnf
```

Within the [mysqld] section add the following lines:

```
innodb_file_per_table=1
lower_case_table_names=0
```

Restart mariadb service

```
# systemctl restart mariadb
```

Step 4: Install and Configure LibreNMS

Add librenms user:

```
# useradd librenms -d /opt/librenms -M -r -s "$(which bash)"
# usermod -a -G librenms apache
```

Clone LibreNMS project from Github

```
# cd /opt
# git clone https://github.com/librenms/librenms.git
# chown librenms:librenms -R /opt/librenms
```

Install PHP dependencies

```
# cd /opt/librenms
# su - librenms
$ ./scripts/composer_wrapper.php install --no-dev
$ exit
```

Configure PHP-FPM

```
cp /etc/php-fpm.d/www.conf /etc/php-fpm.d/librenms.conf
vi /etc/php-fpm.d/librenms.conf
```

Change [**www**] to [**librenms**]:

```
[librenms]
```

Change user and group to "librenms":

```
user = librenms
group = librenms
```

Change listen to a unique name:

```
listen = /run/php-fpm/librenms.sock
```

Configure Web Server

Create the librenms.conf:

```
vi /etc/httpd/conf.d/librenms.conf
```

Add the following config, edit `ServerName` as required:

```
Alias /librenms /opt/librenms/html/
<VirtualHost *:80>
  DocumentRoot /opt/librenms/html/
  ServerName  librenms.mahedi.me
```

```
  AllowEncodedSlashes NoDecode
  <Directory "/opt/librenms/html/">
    Require all granted
    AllowOverride All
    Options FollowSymLinks MultiViews
  </Directory>

  # Enable http authorization headers
  <IfModule setenvif_module>
    SetEnvIfNoCase ^Authorization$ "(.+)" HTTP_AUTHORIZATION=$1
  </IfModule>

  <FilesMatch ".+\.php$">
    SetHandler "proxy:unix:/run/php-fpm/librenms.sock|fcgi://localhost"
  </FilesMatch>
</VirtualHost>
```

Restart Apache and PHP-Fpm

```
# systemctl restart httpd php-fpm
```

Enable lnms command completion

This feature grants you the opportunity to use tab for completion on lnms commands as you would for normal linux commands.

```
ln -s /opt/librenms/lnms /usr/bin/lnms
cp /opt/librenms/misc/lnms-completion.bash /etc/bash_completion.d/
```

Configure snmpd

```
cp /opt/librenms/snmpd.conf.example /etc/snmp/snmpd.conf
vi /etc/snmp/snmpd.conf
```

Edit the text which says RANDOMSTRINGGOESHERE and set your own community string.

```
curl -o /usr/bin/distro https://raw.githubusercontent.com/librenms/librenms-
  agent/master/snmp/distro
chmod +x /usr/bin/distro
systemctl enable snmpd
systemctl restart snmpd
```

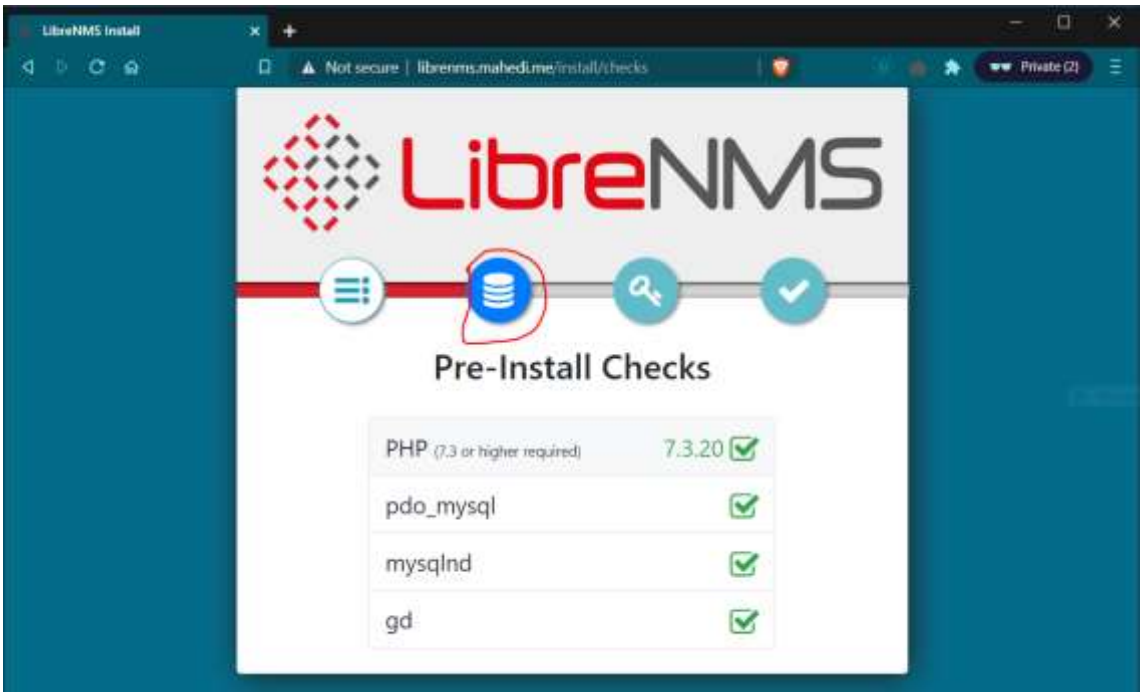
Cron job

```
cp /opt/librenms/librenms.nonroot.cron /etc/cron.d/librenms
```

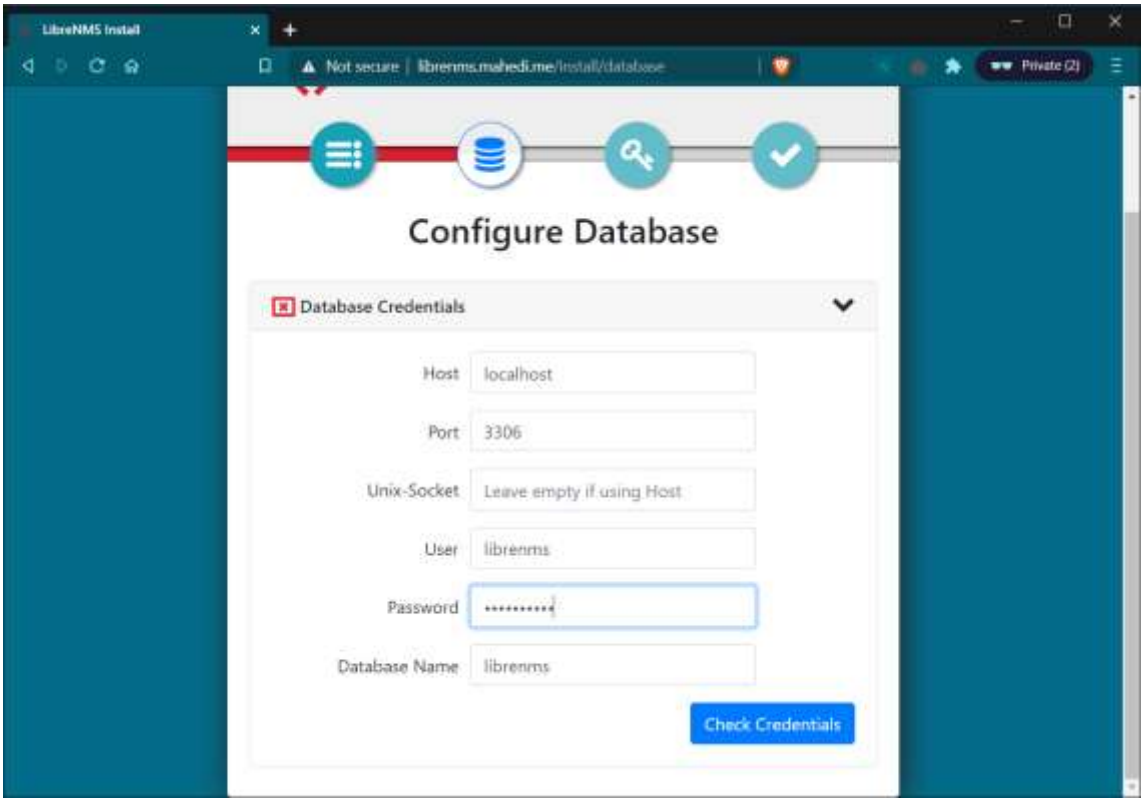
Start Installation:

Open Browser and type the url: http://librenms.mahedi.me/install/

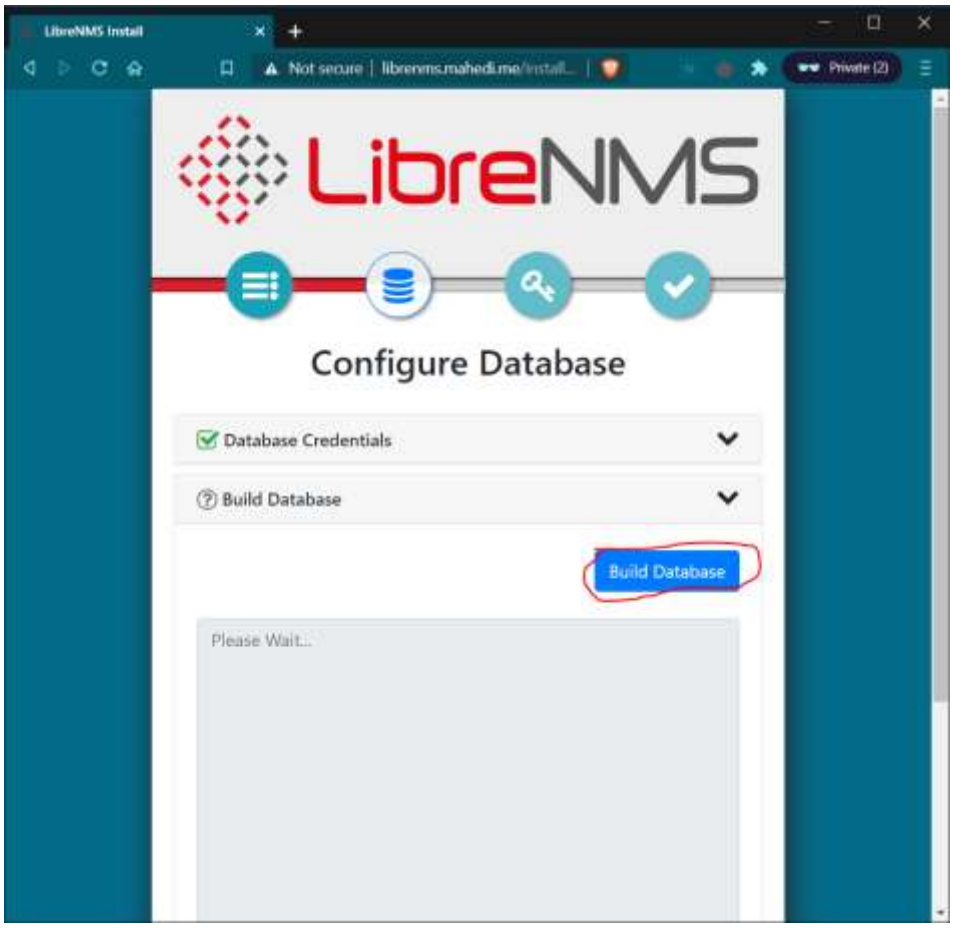
From the installation window click on “Configure Database” icon:



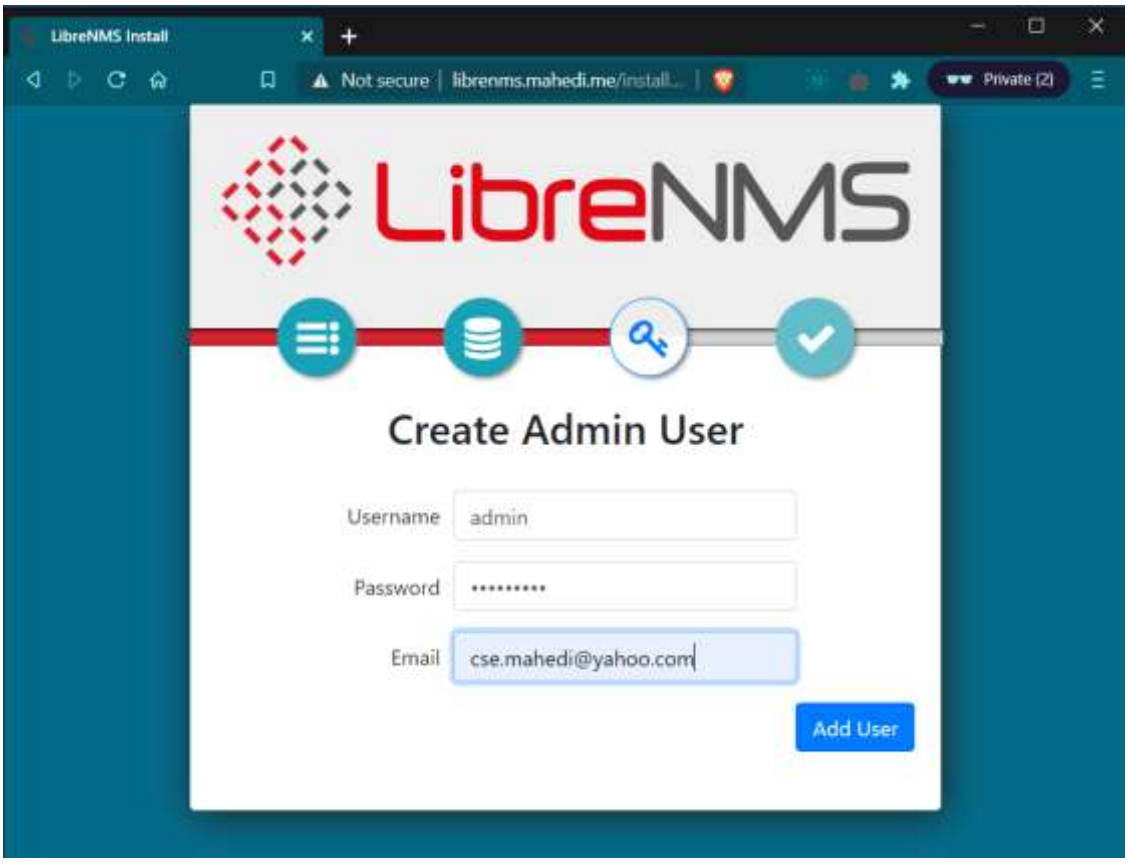
Provide database credentials and click on “Check Credentials”:



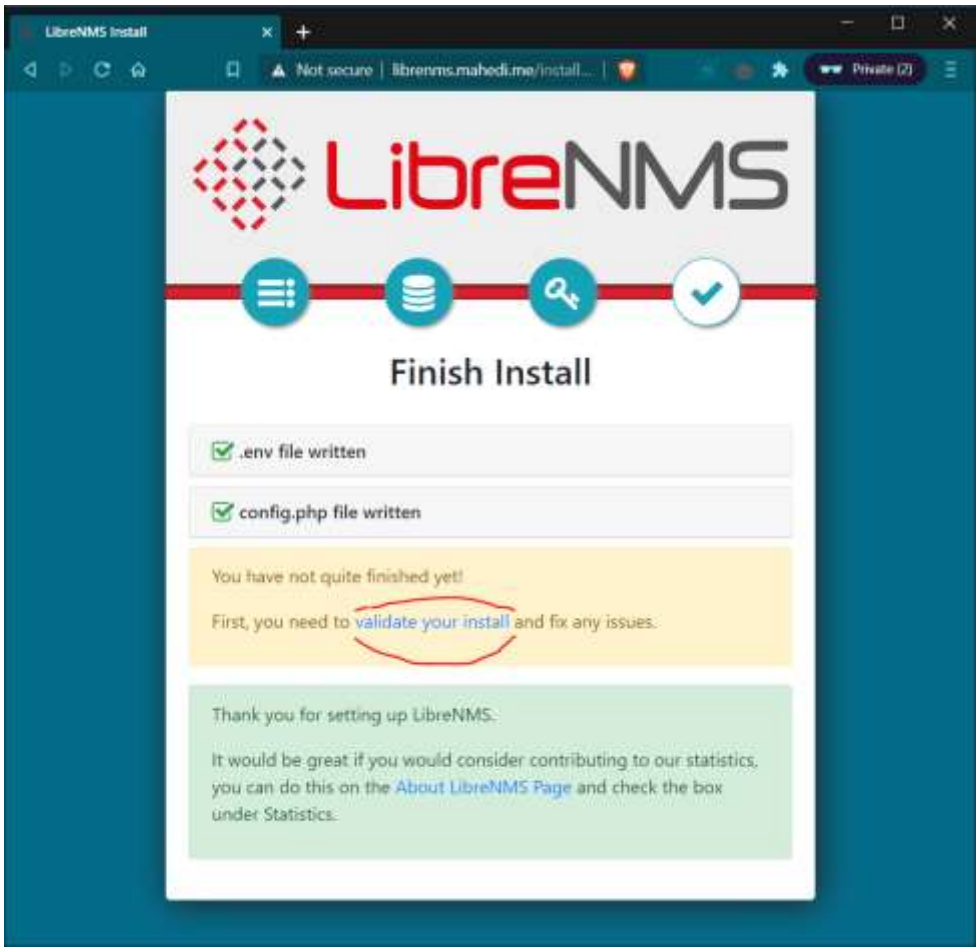
If credential is ok then click on “Build Database”:



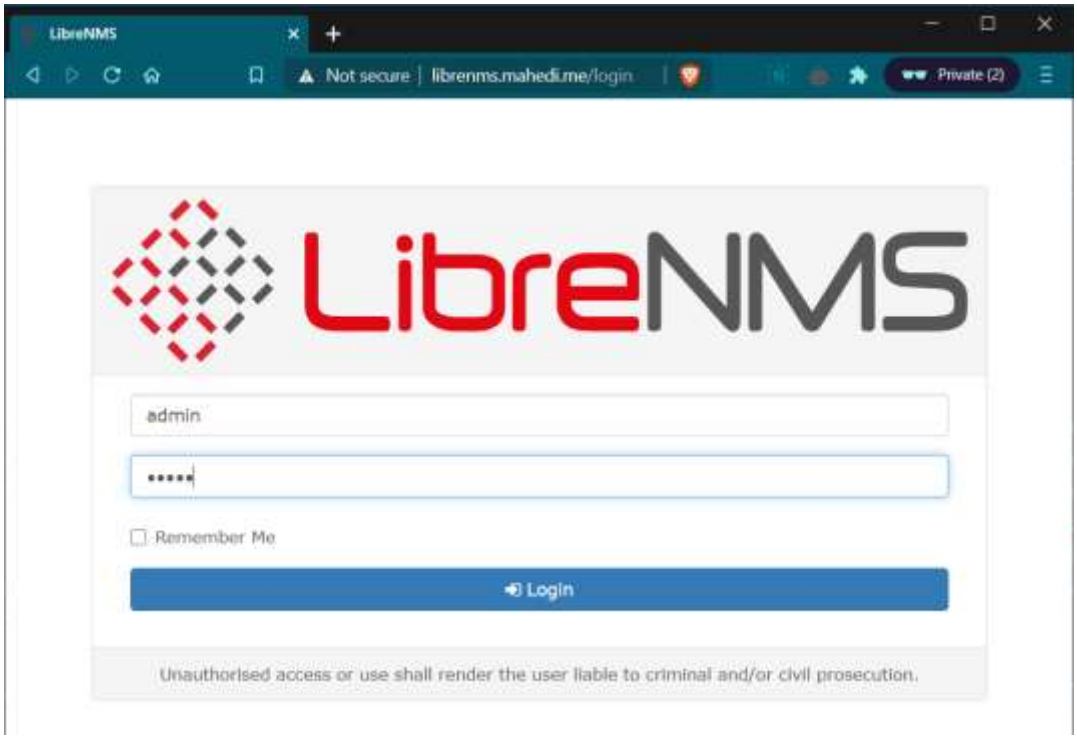
Now Create an Admin account:



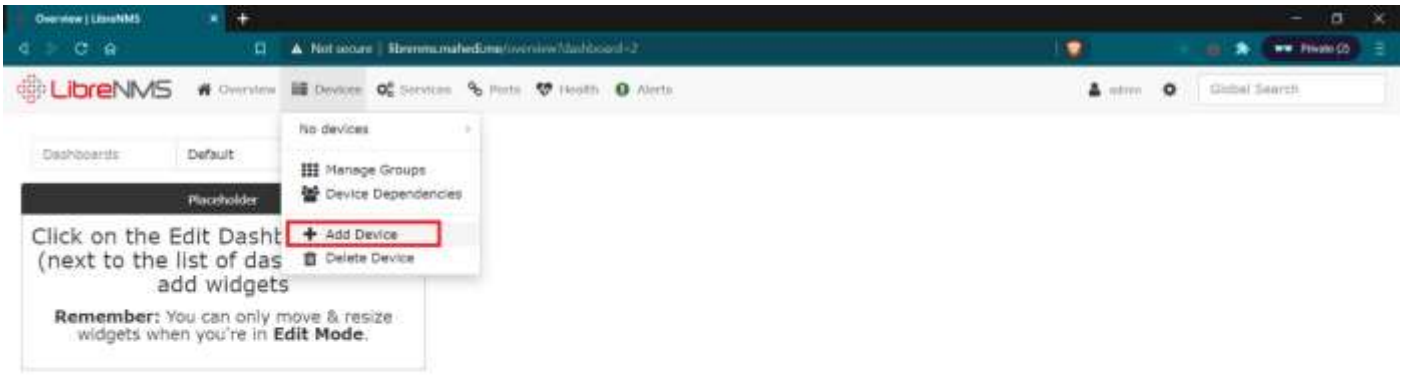
Finally Validate the Installation:



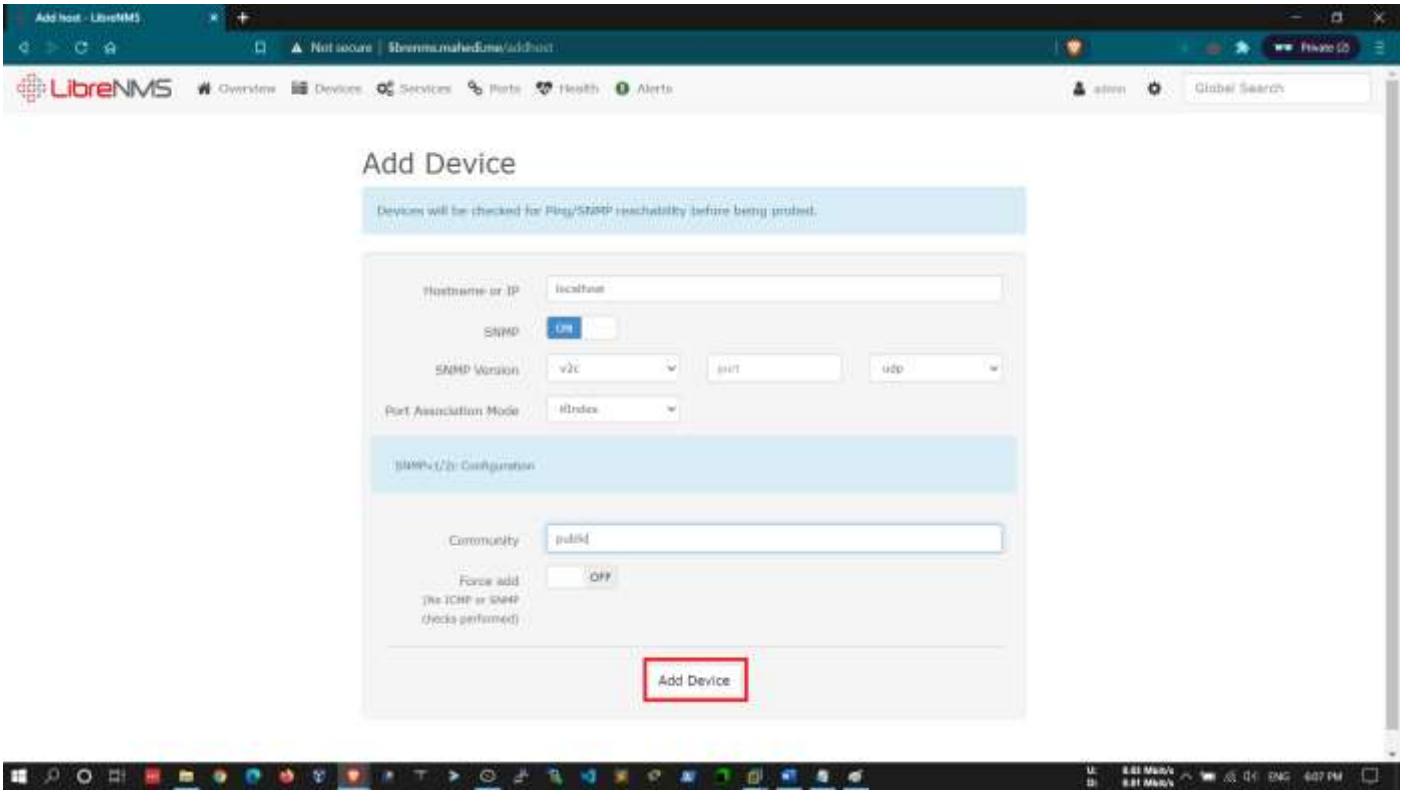
If everything is ok then redirected to login screen:



Finally, Dashboard is showing and Add device from device menu:



Adding Devices:



Device Details: